Building an Application for Learning the Finger Alphabet of Swiss German Sign Language through Use of the Kinect

Nguyen, Phuoc Loc ; Falk, Vivienne ; Ebling, Sarah

Abstract: We developed an application for learning the finger alphabet of Swiss German Sign Language. It consists of a user interface and a recognition algorithm including the Kinect sensor. The official Kinect Software Development Kit (SDK) does not recognize fingertips. We extended it with an existing algorithm.

DOI: https://doi.org/10.1007/978-3-319-08599-9_61

Building an Application for Learning the Finger Alphabet of Swiss German Sign Language through Use of the Kinect

Phuoc Loc Nguyen, Vivienne Falk, and Sarah Ebling

Institute of Computational Linguistics, Zurich, Switzerland
{phuocloc.nguyen,vivienne.falk}@uzh.ch,
ebling@cl.uzh.ch

Abstract. We developed an application for learning the finger alphabet of Swiss German Sign Language. It consists of a user interface and a recognition algorithm including the Kinect sensor. The official Kinect Software Development Kit (SDK) does not recognize fingertips. We extended it with an existing algorithm.

Keywords: Sign language, Swiss German Sign Language, Finger Alphabet, Kinect, Learning Environment.

1 Introduction

Swiss German Sign Language (*Deutschschweizerische Gebärdensprache*, DSGS) is the sign language of the German-speaking area of Switzerland. We developed an application for learning the finger alphabet of this language. It consists of a user interface and a recognition algorithm including the Kinect sensor. The official Kinect Software Development Kit (SDK) does not recognize fingers. We extended it with an existing algorithm to recognize the palm of the hand, the contour of the hand, and the fingertips.

2 Automatic Sign Language Recognition through the Kinect: Related Work

The aim of the DictaSign project [2] was to make online communication easier for Deaf people by means of sign language avatars. During the project, three prototypes were developed: a search-by-example application, a sign language translator, and a sign language wiki. For the search-by-example application, the user signs a phrase in front of the Kinect, the sign sequence is recognized, and possible matches to the input are presented. The user can select the appropriate phrase or add a new phrase. This application was trained for German Sign Language and Greek Sign Language. The sign language translator works similarly: The user signs a phrase which is recognized by the computer with the help of the Kinect. Subsequently, the recognized phrase is shown in four other sign languages. The sign language wiki has the same functionality as a spoken language

K. Miesenberger et al. (Eds.): ICCHP 2014, Part II, LNCS 8548, pp. 404–407, 2014.

wiki, with the difference that here, an avatar presents the content. [1] used the Kinect to recognize Chinese Sign Language.

3 User Interface

To create the learning interface for our application, we used Microsoft's Visual Studio Express 2010. The interface consists of four windows: a main window named *Gebärdensprach-Lernumgebung* ('Sign language learning environment') and three subwindows named *Wissenswertes* ('Useful to know'), *Dein Name* ('Your name'), and *Fingeralphabet* ('Finger alphabet'). The subwindow *Wissenswertes* conveys general information about sign languages, in particular, about Swiss German Sign Language. In the subwindow *Dein Name* ('Your name'), the user can enter a name or word. Upon pressing a button, the corresponding finger signs in Swiss German Sign Language appear. For images of the hand shapes, we relied on a freeware font package[1].

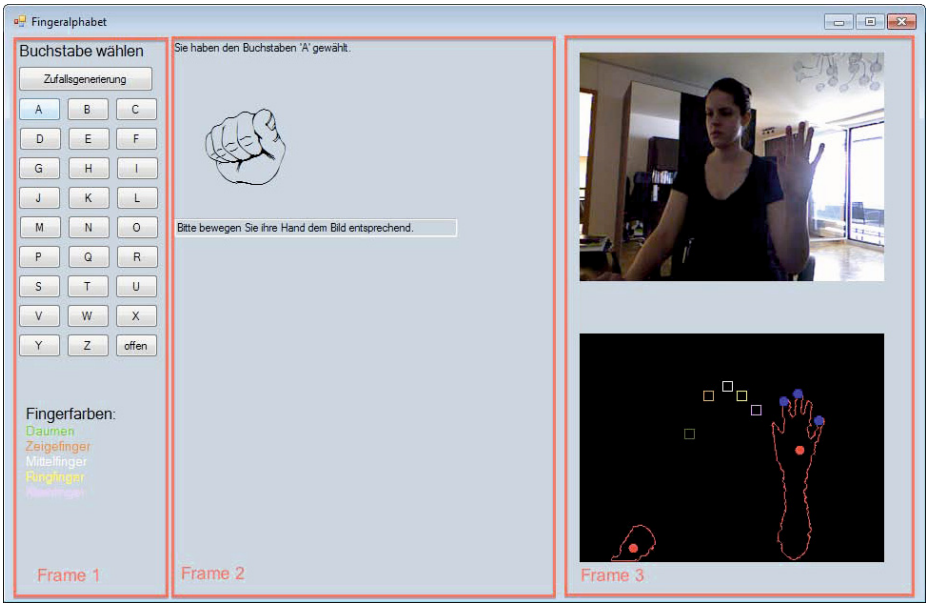

Fig. 1. Window *Fingeralphabet* ('Finger alphabet')

The subwindow *Fingeralphabet* ('Finger alphabet', Figure 1) is where the recognition through the Kinect takes place. In Frame 1, the user can choose a letter by pressing the appropriate button. Alternatively, he or she can generate a random letter. Following the selection of a letter, the appropriate finger sign and a text field appear in Frame 2. In this frame, instructions on how to perform

[1] `http://www.fontspace.com/tanja-schulz/zoefingerabc-regular`

the sign are also given. The Kinect camera creates two images in Frame 3: The upper image shows the user in 2D (color view), the lower is a live tracker of the hand (depth view).

4 Automatic Recognition of Swiss German Sign Language Using the Kinect

The fact that the Kinect is able to provide depth and color data simultaneously made it one of the most important milestones in sign language recognition. The Kinect is capable of obtaining information in a resolution of 640x480 with 30 frames per second. We used the Xbox version without near mode.[2] The Kinect itself can recognize up to six skeletons, but not in detail. The official SDK has no function to recognize fingers. We therefore extended it with the algorithm of [3] to recognize the palm of the hand, the contour of the hand, and the fingertips.[3] The algorithm is divided into eight steps:

1. Define the nearest and farthest point for the Kinect to recognize
2. Decrease noise by applying one or two morphological transformations: dilation and/or erosion
3. Classify a pixel as contour pixel or inside pixel
4. Distinguish hands and calculate their contour
5. Identify pixels inside the hand
6. Find the center of the palm
7. Find the fingertips
8. Allocate points in a 3D space

Following these steps, we have a complete hand detected and are able to use that information: The fingertips are stored in a list and are colored in blue, while the contours and the center of the palm are colored in red, which can be seen in the depth stream in our interface (lower image of Frame 3 in Figure 1). We introduced anchor points in the shape of little squares to guide the user in the process of positioning the fingertips.

The algorithm of [3] provided us with a good approach to recognize hands, but it does not easily recognize finger signs. In particular, there are signs with no extended fingers (e.g., *E*, *M*, *N*, or *S* in the finger alphabet of Swiss German Sign Language) or signs with movements (letters *J* and *Z*), both of which the algorithm cannot handle. There is also no way to distinguish fingertips: The algorithm generates a list of fingertips from left to right. In cases where one or more fingers are not spread out, the list is not complete and the indices for the fingers would not be correct anymore. In other words, we cannot determine whether a finger is in its designated square. Therefore, we just tested whether a fingertip was inside a square at all. If this was the case, the color of the fingertip changed from blue to green.

[2] The near mode allows the device to get reasonable values in a short distance of 40 to 300cm instead of 80 to 400cm.

[3] `http://frantracerkinectft.codeplex.com/`

5 Outlook

The steps to recognize the hand and its fingertips outlined in Section 3 are quite performance-heavy. As our next step, we will improve this part. After that, we will work on improving the actual recognition. The algorithm we applied looks for a specific angle to identify fingertips; if the fingertips are inside the palm, the situation is as if there were no fingers. We plan to include a second camera to provide information from an additional angle. At the same time, we will try to better exploit the 3D information provided by the Kinect. We will perform our experiments with the new version of the Kinect that was recently released.

A more distant goal is to extend our application so that full signed sentences may be recognized. For this, it will be necessary to combine hand/finger recognition with face and body recognition.

References

1. Chai, X., Li, G., Chen, X., Zhou, M., Wu, G., Li, H.: VisualComm: A tool to support communication between deaf and hearing persons with the Kinect. In: Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS 2013, pp. 76:1–76:2. ACM, New York (2013), http://doi.acm.org/10.1145/2513383.2513398
2. Efthimiou, E., Fotinea, S.E., Hanke, T., Glauert, J., Bowden, R., Braffort, A., Collet, C., Maragos, P., Lefebvre-Albaret, F.: Sign language technologies and resources of the Dicta-Sign project. In: Proceedings of the 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, LREC 2012, Istanbul, Turkey, pp. 37–45 (2012)
3. Trapero Cerezo, F.: 3D Hand and Finger Recognition using Kinect. Tech. rep., Universidad de Granada (UGR), Spain (2012)